Subsystem Design

Scanning Comparator

1 Description

This subsystem demonstrates how to represent multiple comparators with a single integrated comparator and software in an MSPM0 microcontroller. The process allows the designer to maximize the comparator function and utilize more theoretical comparators than are physically on the device. This example specifically cycles through three different comparator configurations and input pins while setting three output pins with the results as shown in Figure 1-1.

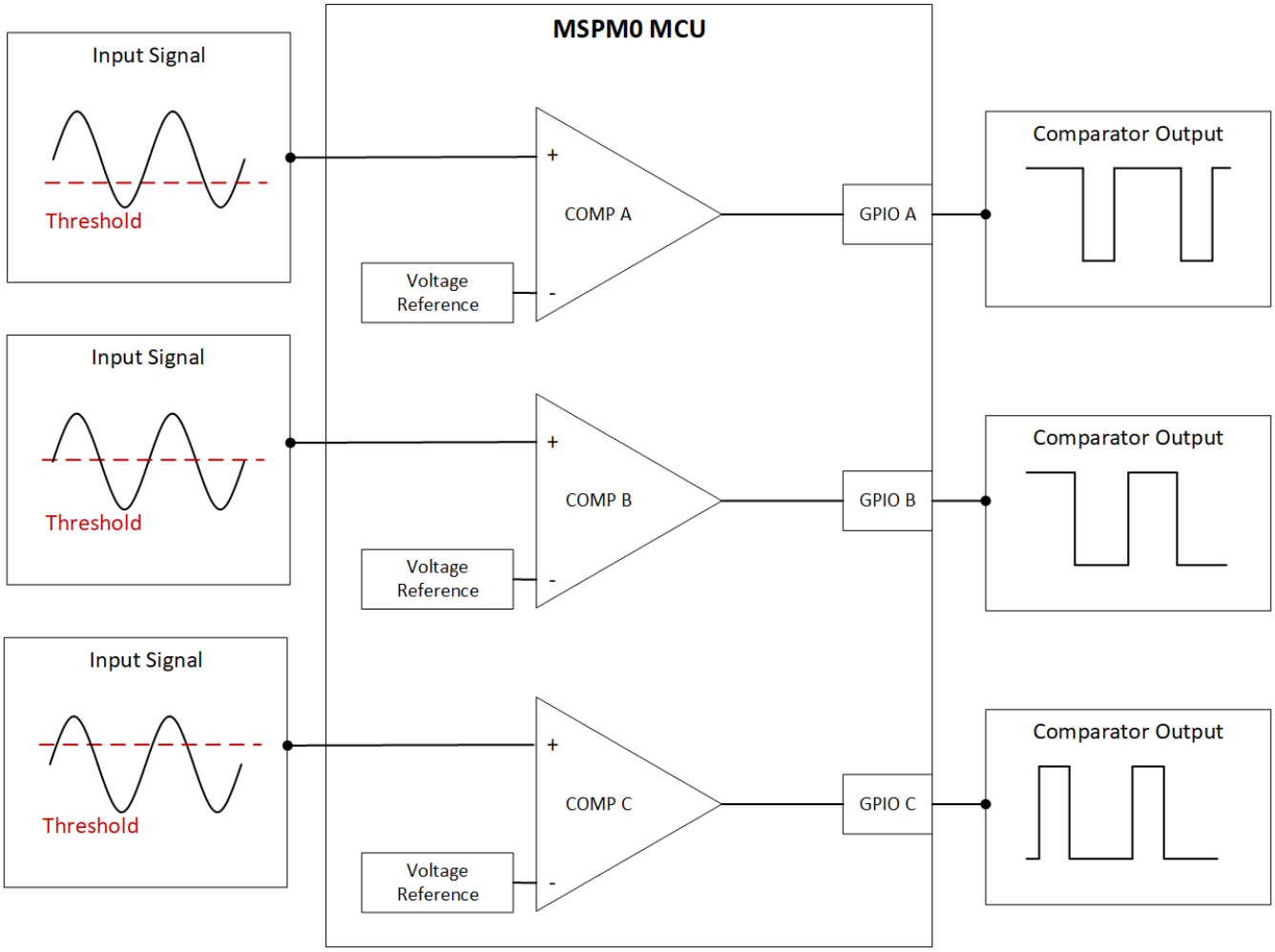

Figure 1-1. Theoretical Function of Scanning Comparator Subsystem

Utilizing the customizable IO MUXing for the MSPM0 comparator, this example enables multiple signal inputs for the same comparator. The three signal inputs for this example are on the COMP_IN0+, COMP_IN0-, and COMP_IN1- pins as shown in Figure 1-2.

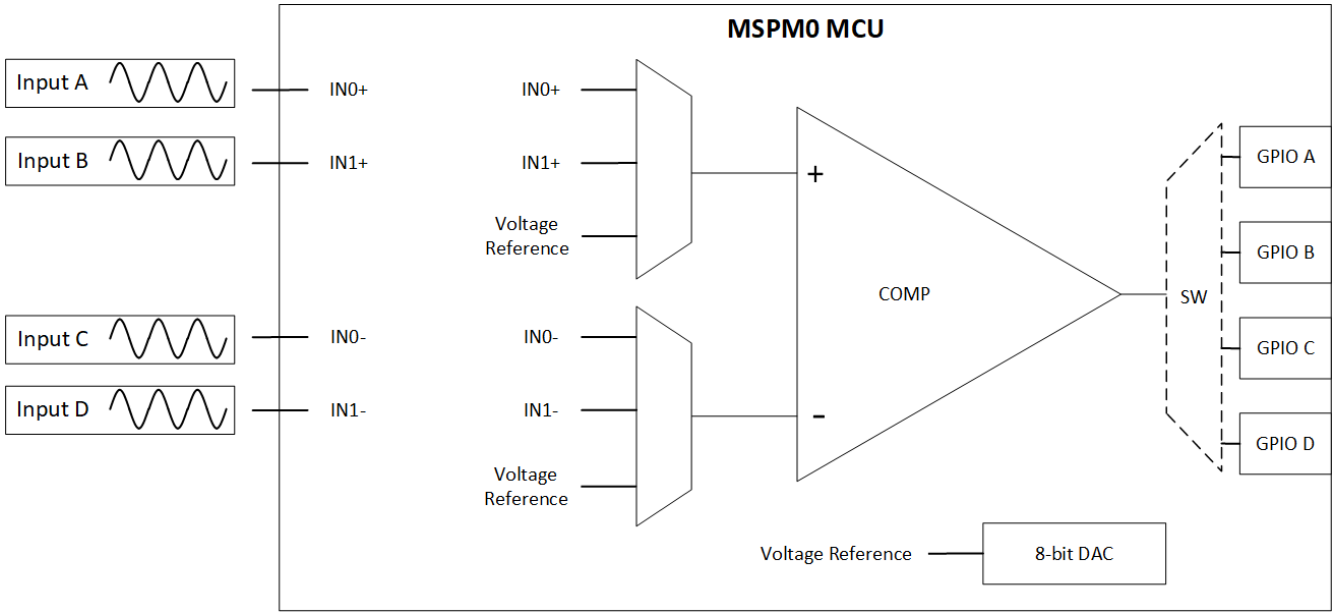

Figure 1-2. Comparator Input and Output MUXes

2 Required Peripherals

Table 2-1 describes the *required* integrated COMP and GPIOs.

Table 2-1. Required Peripherals

Peripheral Used	Notes
Comparator	Called COMP_INST in code (Includes 8-bit reference DAC)
GPIO	The three GPIOs are referred to as pins A, B, and C.

3 Compatible Devices

Based on the requirements shown in Table 2-1, this example is compatible with the devices shown in Table 3-1. The corresponding EVM can be used for prototyping.

Table 3-1. Compatible Devices

Compatible Devices	EVM	Hardware COMP	Maximum COMP Inputs
MSPM0L13xx	LP-MSPM0L1306	1	4
MSPM0Lx22x	LP-MSPM0L2228	1	4
MSPM0Gx5xx	LP-MSPM0G3507	3	17

4 Design Steps

1. Determine multiple configurations for the comparator including operating mode, channel inputs, and voltage reference based on design requirements.
2. Generate comparator configuration code utilizing SysConfig.
3. Configure the required GPIOs in SysConfig.
4. Create separate functions for each comparator configuration from Steps 1–2.
5. Write application code to call each configuration setting, delay for settling time, and assign the result to the corresponding IO pin. See Figure 6-1 for an overview of the software.

5 Design Considerations

1. Settling time: After updating the configuration of the comparator, the application code requires a delay to allow for the enable time, DAC settling time, and propagation delay before reading the result. When setting the delay in the application code, refer to the comparator specifications section of the respective MSPM0 data sheet.
2. Operating mode: The comparator has both a high-speed and a low-power mode. The high-speed mode consumes more current, but decreases the time between comparator readings. The low-power mode requires longer delays between readings, but decreases the current consumption of the device. For reference, this example use the high-speed mode.
3. Response time: As the subsystem cycles through multiple comparator configurations, this process increases the maximum comparator response time. The maximum response time is a factor of the settling time delay multiplied by the number of emulated comparator configurations. Standard response time = x; Emulated response time = delay × emulated comparator (45µs)

6 Software Flow Chart

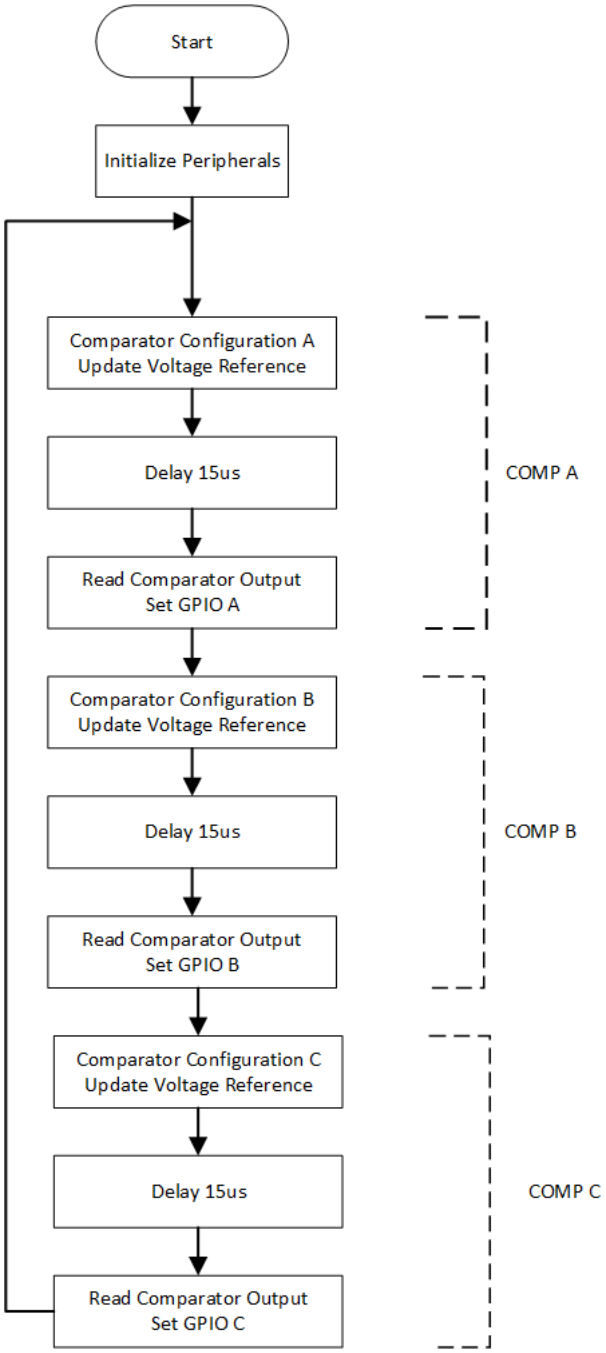

Figure 6-1. Application Software Flow Chart

7 Application Code

The application code cycles through three different comparator configurations by calling the three functions: `update_comp_configA()`, `update_comp_configB()`, and `update_comp_configC()`. Each time after reconfiguring the comparator, the application code delays 15µs for the propagation and settling delay before reading the comparator output and setting the respective GPIO.

```
int main(void)
{   //initialization
    SYSCFG_DL_init();
    DL_COMP_enable(COMP_INST);
    DL_SYSCTL_enableSleepOnExit();

    while (1) {

        //0.5V reference
        update_comp_configA();
        delay_cycles(480); //15us delay for comp stabilization
        if (DL_COMP_getComparatorOutput(COMP_INST) == 1){
                    DL_GPIO_setPins(COMP_OUTPUT_PORT,COMP_OUTPUT_A_PIN); //set GPIO high
                }else{
                    DL_GPIO_clearPins(COMP_OUTPUT_PORT,COMP_OUTPUT_A_PIN); //set GPIO low
                }

        //1.0V reference
        update_comp_configB();
        delay_cycles(480); //15us delay for comp stabilization
        if (DL_COMP_getComparatorOutput(COMP_INST) == 1){
                    DL_GPIO_setPins(COMP_OUTPUT_PORT,COMP_OUTPUT_B_PIN); //set GPIO high
                }else{
                    DL_GPIO_clearPins(COMP_OUTPUT_PORT,COMP_OUTPUT_B_PIN); //set GPIO low
                }
        //1.5V reference
        update_comp_configC();
        delay_cycles(480); //15us delay for comp stabilization
        if (DL_COMP_getComparatorOutput(COMP_INST) == 1){
                    DL_GPIO_setPins(COMP_OUTPUT_PORT,COMP_OUTPUT_C_PIN); //set GPIO high
                }else{
                    DL_GPIO_clearPins(COMP_OUTPUT_PORT,COMP_OUTPUT_C_PIN);//set GPIO low
                }
    }
}
```

Figure 7-1. Scanning Comparator Main.C

8 Results

Figure 8-1 shows the results of the scanning comparator subsystem example. Emulated comparators A, B, and C had reference voltages set to 0.5V, 1.0V, and 1.5V, respectively. The same input signal was measured on each of the three emulated comparators.

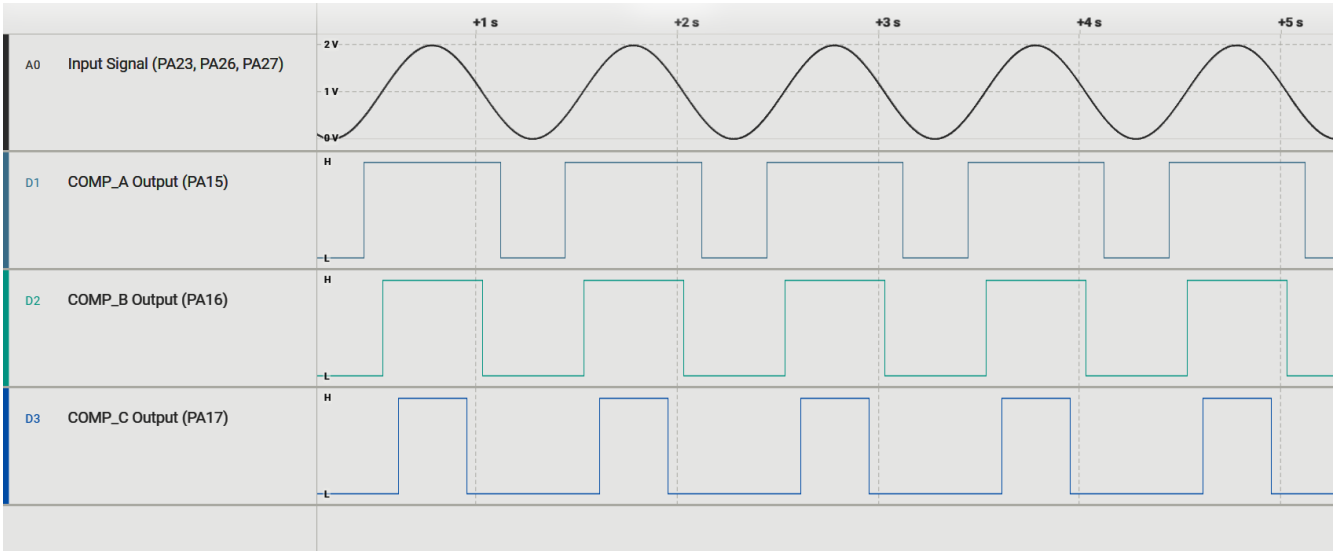

Figure 8-1. Results

The scope reading demonstrates that one physical comparator was able to mimic three comparators running at the same time. This example code can be edited to fit different comparator numbers and configurations by changing the functions within `comp_hal.c`.

9 Additional Resources

- Texas Instruments, Download the MSPM0 SDK
- Texas Instruments, Learn more about SysConfig
- Texas Instruments, MSPM0L LaunchPad™
- Texas Instruments, MSPM0G LaunchPad™
- Texas Instruments, MSPM0 Academy

10 E2E

See TI's E2E™ support forums to view discussions and post new threads to get technical support for utilizing MSPM0 devices in designs.

Trademarks

LaunchPad™ and E2E™ are trademarks of Texas Instruments.
All trademarks are the property of their respective owners.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265